REMARKS

In the Office Action, the Examiner indicated that claims 1-17 are pending in the application and the Examiner rejected all claims. The rejections are respectfully traversed below.

Claim Amendments

Applicant has amended the claims to present independent claims of narrower scope. Specifically, the content of claims 2-4 has been added to claim 1, and the content of claims 11 and 15 has been added to claim 7.

Rejections under 35 U.S.C. §103

On page 2 of the Office Action, the Examiner rejected claims 1-17 under 35 U.S.C. §103 as being unpatentable over U.S. Patent Publication No. 20020170048 to Zgarba in view of U.S. Patent No. 4,931,928 to Greenfeld.

The Present Invention

The present invention is an asset locator (search engine) for locating software assets, code assets and the like that are stored in code repositories used by software designers. It provides the capability for the gathering of information about assets contained in the code repositories and the capturing of the gathered information in a database that can be used for the conducting of subsequent searches. The present invention has particular application in a software-development environment where the stored code assets may number in the millions and may be written in diverse languages such as, for example, Java, C/C++, COBOL, HTML, and/or XML.

A crawl process is performed on a storage device on which assets are stored to identify the assets. Asset-specific parameters related to the stored assets are identified, and the assets are then analyzed based upon these parameters. Textual and semantic information is extracted from the stored assets and then the extracted textual and semantic information is stored and indexed for retrieval.

In a preferred embodiment, a series of data analyzers that are specific to each type of data contained in the code repositories (e.g., a Java analyzer, a C/C++ analyzer, a COBOL analyzer, an HTML analyzer, and/or an XML analyzer) are integrated into the system so that they can be used to search the code repositories using particular attributes specific to the semantics of a particular language used to code the asset. In another preferred embodiment, the repositories are crawled automatically according to a schedule defined by the user, and the results of the crawling are stored in a database. Ordinary keyword searching can then be used with the system, either independently or combined with the attribute-specific semantic searching, to search the database.

U.S. Patent Publication No. 20020170048 to Zgarba et al.

U.S. Patent Publication No. 20020170048 to Zgarba et al. ("Zgarba") teaches a method of forward engineering code previously reverse engineered into a software model.

Elements from the original source code represented by the model are placed in a meta-model, and compared to a similar meta-model of the software model. The Examiner asserts that paragraphs 51-53 of Zgarba teach various elements of claims 1-17.

U.S. Patent No. 4,931,928 to Greenfeld

U.S. Patent No. 4,931,928 to Greenfeld ("Greenfeld") teaches an apparatus used for analyzing source code. An analysis member extracts programming semantics information from input source code. Greenfeld is interested in analyzing the source code for the purposes of reverse engineering and thus is only interested in the semantic elements of the source code and has no need or motivation to analyze the text, such as, the text appearing in comments.

The Examiner has not Established a *prima facie* Case of Obviousness

To support a rejection under 35 U.S.C. §103, a reason, suggestion, or motivation to lead an inventor to combine two or more references must be found. *Pro-Mold and Tool Co. v. Great Lakes Plastics Inc.*, 37 U.S.P.Q.2d 1627, 1629 (Fed.Cir. 1996) (see also MPEP 2143). The Examiner has not met this burden, as set forth below.

The Examiner asserts that claims 1-17 are unpatentable under 35 U.S.C. §103(a) based on Zgarba et al. in view of Greenfeld. As noted above, applicant has amended the independent claims to claim narrower aspects of the present invention in the independent claims (and thus in all of the dependent claims as well). Specifically, each of the independent claims recite not only that both textual and semantic analysis is performed, but that this dual extraction of information is performed across code assets of diverse types. The amendments to the claims are based on already claimed subject matter, i.e., some of the narrower limitations of dependent claims have been moved into the independent claims.

Neither Zgarba nor Greenfeld teach or suggest (a) the ability or need to extract and index both textual and semantic information from code assets, nor (b) performing such dual-type analysis across diverse types of code assets. This capability, which is provided only by the present invention, allows significantly more precision to be used when searching code assets, as established in the examples given in the specification.

Regarding the Examiner's assertions as to the teachings of Zgarba in paragraphs 51-53, applicant believes that these paragraphs do not contain the teachings asserted by the Examiner. For example, the Examiner asserts that Zgarba teaches "performing a crawl process on said storage device to identify stored assets" and cites page 4, paragraph 0052, of Zgarba. This paragraph is repeated below in its entirety.

> "[0052] The operation of the meta-model generation and the nature of the meta-models generated will depend on the structure of the programming language and the constructs therein, as well as the constructs used in the software model. It must be possible to search through the elements in the source-code meta-model 80 and identify corresponding elements from the software model 2 and the source code 4. For example, different classes, variables and other independent structures in the source code such as STRUCT and UNION structures could be stored in a linked list. Elements owned by these structures could be placed in a linked list pointed to by the appropriate element in the main linked list. Searching for elements is then easily accomplished by passing down the linked list, finding the appropriate elements, and if

> necessary then looking through the appended linked list associated with that element."

Nothing in this paragraph remotely suggests crawling a repository, let alone crawling a repository of code assets of diverse types as is currently claimed. As set forth in previous arguments, the term "crawling" in computer parlance has a well-defined and well known meaning, and it cannot be considered analogous to the simple "searching" referenced in paragraph 0052 of Zgarba.

Similarly, the Examiner asserts that paragraph 0053 of Zgarba teaches the analysis of stored assets based on identified asset-specific parameters. Following is paragraph 0053 from Zgarba, repeated in its entirety.

> "[0053] Importantly, each element in the generic meta-model generated from the source code has associated with it a line number 84 and a "change state "86. The line number corresponds to the line number of the element in question in the original source code. In alternative embodiments of the invention, the actual start and finish points in the code associated with any particular element could be stored in the software model, rather than just the line number, so that multiple elements on one line can be independently merged. For example, in the example meta-model generated from the source code shown in FIG. 9A, the line numbers 84 are shown in the center column."

Nowhere in this paragraph (nor anywhere else in Zgarba) is there a teaching or suggestion of the analysis and storing of assets based on identified asset-specific parameters as set forth in the rest of the claim.

As noted above, Greenfeld makes no mention nor no suggestion of text searching and is solely interested and capable of doing semantic searching for the purpose of reverse engineering. While this may be a useful tool to an engineer attempting to reverse engineer code, it cannot and does not perform the claimed elements.

In short, the claimed invention is neither taught nor suggested by Zgarba or Greenfeld, either alone or in combination. Accordingly, the Examiner is respectfully requested to reconsider and withdraw the rejection of claims 1, 5-10, 12-14 and 16-17 under 35 U.S.C. §103.

Conclusion

The rejection of claims 1-17 have been traversed. Accordingly, reconsideration of the present application, and withdrawal of the rejections on the grounds of 35 U.S.C. §103 is respectfully requested.

A Petition for extending the time to respond to the Examiner's Action two months is enclosed in duplicate. The Commissioner is hereby authorized to charge any additional fees or credit any overpayment associated with this communication to Deposit Account No. 19-5425.

Respectfully submitted,

APRIL 15, 2005
Date

Mark D. Simpson, Esq.
Registration No. 32,942
SYNNESTVEDT & LECHNER LLP
Suite 2600 Aramark Tower
1101 Market Street
Philadelphia, PA 19107
Telephone: (215) 923-4466
Facsimile: (215) 923-2189

M:\MSIMPSON\CLIENTS\IBM RALEIGH RSW\23546\PATOFF\REPLY TO ACTION OF 12022004.DOC